

REMARKS/ARGUMENTS

Claims 1-39 are pending in the application. Claims 1, 6, 8, 14, 19, 21, 27, 32, and 34 have been amended. Reconsideration is respectfully requested. Applicants submit that the pending claims 1-39 are patentable over the art of record and allowance is respectfully requested of claims 1-39.

Claims 1-39 are rejected under 35 U.S.C. 102(b) as being anticipated by Krishna (U.S. Patent No. 5,412,804). Applicants respectfully traverse.

Anticipation requires that the identical invention must be shown in a single reference in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

Claims 1, 14, and 27 describe, under control of an iterator function processor, when an iterator function included in a statement is invoked, obtaining one or more predicates included in the statement, applying the one or more predicates to a row of data, if applying the one or more predicates results in a match, returning the row of data, and if applying the one or more predicates does not result in a match, searching for another row of data for which application of the one or more predicates results in a match.

On the other hand, as described in Applicants' Background of the Invention, a conventional iterator function receives a set of arguments and *returns a table* to the SQL statement that invokes the function (e.g., Specification, page 1, paragraph 4). In particular, the *iterator function creates a virtual table with a result set, and then the qualification (i.e., predicate) is applied to the virtual table to filter rows* of data in the virtual table (e.g., Specification, page 3, paragraph 9). In many cases, only a small percentage of rows of data in the virtual table remain after the qualification is applied, but, unfortunately, because the qualification is applied after the rows of data are retrieved for the result set, many rows of data are unnecessarily retrieved for the virtual table (e.g., Specification, page 3, paragraph 9). For example, an application program submitting SQL statement (2) is interested in articles with a score (assigned by the `text_search()` iterator function) greater than or equal to 0.9, but, because the `text_search()` iterator function is not aware of the score qualification, the `text_search()` iterator function returns a virtual table with all articles that have "Bush" in the same sentence as

"recession," without applying the predicate for the score (e.g., Specification, page 3, paragraphs 10-11).

To avoid this problem, under control of an iterator function processor, when an iterator function included in a statement is invoked, one or more predicates included in the statement are obtained and the one or more predicates are applied to a row of data before the row of data is returned.

The Krishna patent does not address how iterator functions are processed. Instead, the Krishna patent describes that a query is converted or parsed into a more fixed format for processing by the query optimizer and execution by the retrieval system (Col. 6, lines 26-30).

Thus, claims 1, 14, and 27 are not anticipated by the Krishna patent.

Dependent claims 2-5, 15-18, and 28-31 incorporate the language of independent claims 1, 14, and 27 and add additional novel elements. Therefore, dependent claims 2-5, 15-18, and 28-31 are not anticipated by the Krishna patent for at least the same reasons as were discussed with respect to claims 1, 14, and 27.

Also, claim 2 describes that, under control of an iterator function processor, when an iterator function included in a statement is invoked (from independent claim 1), obtaining the one or more predicates comprises obtaining a qualification descriptor that describes the one or more predicates and one or more functions. The Krishna patent describes a query node (FIG. 4), but this does not teach or suggest that an iterator function processor obtains the claimed qualification description when an iterator function is invoked.

Claim 5 describes that the iterator function is invoked by a data store engine and that the row of data is returned to the data store engine. As explained in Applicants' Background of the Invention, a conventional iterator function receives a set of arguments and *returns a table*. The Krishna patent at Col. 9, lines 39-51, describes aggregate processing. Such aggregate processing does not anticipate the claimed processing of an iterator function.

Claims 6, 19, and 32 describe under control of a data store engine, receiving a statement including an iterator function and one or more predicates, creating a qualification descriptor that describes the one or more predicates and one or more functions that are to be used to evaluate the one or more predicates, and invoking the iterator function one or more times, until receiving a done indicator from the iterator function, wherein each invocation of the iterator function results

in receiving either a row of data for which at least one predicate has been applied or the done indicator. Thus, unlike with a conventional iterator function that returns a table, each invocation of the iterator function returns either a row of data for which at least one predicate has been applied or the done indicator.

The Krishna patent does not address how iterator functions are processed. The Krishna patent at Col. 10, lines 1-5 describes "tuple iteration" that executes the query block by sequentially indexing the rows of the relations in the relation list so that all combinations of rows are indexed. Such tuple iteration in the Krishna patent does not describe invoking the iterator function one or more times, until receiving a done indicator from the iterator function, wherein each invocation of the iterator function results in receiving either a row of data for which at least one predicate has been applied or the done indicator.

Thus, claims 6, 19, and 32 are not anticipated by the Krishna patent.

Dependent claims 7-11, 20-24, 33-37 incorporate the language of independent claims 6, 19, and 32 and add additional novel elements. Therefore, dependent claims 7-11, 20-24, 33-37 are not anticipated by the Krishna patent for at least the same reasons as were discussed with respect to claims 6, 19, and 32.

Also, claim 8 describes that, when the iterator function is invoked, receiving an indication from the iterator function indicating which of the one or more predicates were applied by the iterator function (e.g., Specification, page 13, paragraphs 38-42). The Krishna patent at Col. 10, lines 47-52, describes that the predicate is tested, and FIG. 8, block 172, determines whether a predicate is true, but this does not anticipate receiving an indication of *which of* the one or more predicates were applied by the iterator function.

Moreover, claims 10 and 11 describe applying one or more additional predicates to the received row of data. The Krishna patent does not describe that an iterator function applies one or more predicates, while the data store engine applies one or more additional predicates.

Claims 12, 25, and 38 describe the interaction between a data store engine and an iterator function processor in processing an iteration function. Under control of a data store engine, a statement including an iterator function and one or more predicates is received, a qualification descriptor is created that describes the one or more predicates and one or more functions that are to be used to evaluate the one or more predicates, and the iterator function is invoked. Under

control of an iterator function processor, the qualification descriptor is retrieved, a row of data that matches the qualification in the qualification descriptor is obtained, and the row of data is returned to the data store engine.

Applicants respectfully submit that the Krishna patent does not anticipate the processing by the data store engine and iterator function processor in processing an iterator function in claims 12, 25, and 38.

Dependent claims 13, 26, and 39 incorporate the language of independent claims 12, 25, and 38 and add additional novel elements. Therefore, dependent claims 13, 26, and 39 are not anticipated by the Krishna patent for at least the same reasons as were discussed with respect to claims 12, 25, and 38.

Conclusion

For all the above reasons, Applicants submit that the pending claims 1-39 are patentable over the art of record. Applicants have not added any claims. Nonetheless, should any additional fees be required, please charge Deposit Account No. 09-0460.

The attorney of record invites the Examiner to contact her at (310) 553-7973 if the Examiner believes such contact would advance the prosecution of the case.

Dated: June 23, 2006

By: /Janaki K. Davda

Janaki K. Davda
Registration No. 40,684

Please direct all correspondences to:

David Victor
Konrad Raynes & Victor, LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, CA 90212
Tel: 310-553-7977
Fax: 310-556-7984